The source files for these theorems, named within square brackets above, may be obtained individually from the directory
ftp://ftp.cli.com/pub/nqthm/nqthm-1992/examples/
or altogether in the single file
ftp://ftp.cli.com/pub/nqthm/nqthm-1992/1995-examples.tar.Z.

Also included on the tar file are new "driver" files for doing a replay of all the examples under Nqthm-1992, both these new examples and those previously distributed with Nqthm-1992. A Gnu Emacs TAGS file for all the event commands in all the examples is also provided.

For information on obtaining the Nqthm prover itself, see
ftp://ftp.cli.com/pub/nqthm/nqthm-1992/nqthm-1992.announcement.

An Erratum for Some Errata to Automated Theorem Proving Problems

Francis Jeffrey Pelletier
Department of Computing Science, University of Alberta,
Edmonton, Alberta, Canada T6G 2H1. e-mail jeffp@cs.ualberta.ca
Geoff Sutcliffe
Department of Computer Science, James Cook University,
Townsville, Queensland, Australia 4811. e-mail geoff@cs.jcu.edu.au

1 Introduction

In 1986 Pelletier [7] published an annotated list of logic problems, intended as an aid for students, developers, and researchers to test their automated theorem proving (ATP) systems. The 75 problems in the list are subdivided into propositional logic (Problems 1–17), monadic-predicate logic (Problems 18–34), full predicate logic without identity and functions (Problems 35–47), full predicate logic with identity but without functions (Problems 48–55), full predicate logic with identity and arbitrary functions (Problems 56–70), and problems to use in studying computational complexity of ATP systems (Problems 71–75). The problems were chosen partially for their historical interest and partially for their abilities to test different aspects of ATP systems. The problems were also assigned an intuitive "degree of difficulty", relativized to the type of problem. All the problems are presented in a "natural form" (which is here also called the "first-order form" or FOF), and most of them are also given in an equivalent negated conclusion clause normal form (CNF). The CNF versions of the problems are all in the TPTP Problem Library [12, 13], and are thus conveniently available to ATP system developers who use the CNF form.[1]

[1]The TPTP Problem Library can be obtained through the World Wide Web (WWW) URLs:
http://www.cs.jcu.edu.au/ftp/users/GSutcliffe/TPTP.HTML
http://wwwjessen.informatik.tu-muenchen.de/~suttner/tptp.html

2 The Erratum

Shortly after the publication of [8], Art Quaife and John Pollock sent Jeff Pelletier lists of errors occurring in the problems. These errata (and some others) were published in [9]. (Researchers thinking of using the problems should most definitely consult the Errata!) In particular, Problem 62 was "corrected". Since then a few researchers have written to Pelletier questioning the validity of some of the problems (usually not saying whether they were working with the natural FOF or the CNF), but he always maintained that they were valid, since they were provable with THINKER [6, 7], his natural deduction-based ATP system. So he said that these researchers either had an inferior ATP system or were not looking at the Errata. Recently, however, Geoff Alexander reported to Geoff Sutcliffe that he was unable to find a proof of the "corrected" CNF version of Problem 62. This report caused Geoff Sutcliffe to attempt, and fail, to prove it with OTTER [5]. Geoff Alexander then pointed out to Geoff Sutcliffe that the "corrected" CNF version of Problem 62 in [9] does not contain any negative clauses. Geoff used Geoff's insight to convince Jeff that not all was well with even the "corrected" version of Problem 62. Since the "corrected" natural FOF of Problem 62 is provable by THINKER (and by John Pollock's system, Oscar [10, 11]), but the CNF version of the problem is satisfiable, the obvious conclusion is that this CNF version was constructed incorrectly.

There are various reasons why it is not possible to establish exactly went wrong: Pelletier's original clausifier is no longer available, and other clausifiers (e.g., Sutcliffe's and OTTER's) apparently use algorithms different from Pelletier's and from each other, and generate different sets of clauses. In the end it was concluded that the CNF version of Problem 62 in the Errata was generated from the unnegated-conclusion of the natural form. As with the original errata, Pelletier again takes responsibility for this further erratum.

In discovering this erratum in the Errata, yet another flaw was discovered. The natural form of Problem 62 in [9] has an $\rightarrow$ (implication) as the main connective, whereas the original version in [8] has an $\leftrightarrow$ (equivalence). Further, Problem 62 is one of four variations on the same problem, and all the others (Problems 17, 33, 38) have an $\leftrightarrow$ as main connective in their natural forms. It is clearly intended that Problem 62 should also have an $\leftrightarrow$ as its main connective in the natural form. Thus both the CNF and FOF versions of Problem 62 in [9] are incorrect (although the FOF, unlike the CNF, is nonetheless still a theorem).

3 The Correction

The correct version of Problem 62 is (following the format of [8], + is disjunction, & is conjunction, u, v, w, x, y, and z are variables, (Ax) is a universally quantified x, (Ex) is an existentially quantified x, f is a function symbol, P is a predicate letter, a is a constant, and skN are Skolem constants generated in the clausification):

Problem 62

Natural FOF	Negated Conclusion CNF
$(Ax)[(Pa\&(Px \to Pf(x))) \to Pf(f(x))]$	Pa
$\leftrightarrow$	$Px + Pf(f(x)) + \neg Pa$
$(Ax)[(\neg Pa + Px + Pf(f(x)))\&$	$Pf(f(x)) + \neg Pa + \neg Pf(x)$
$(\neg Pa + \neg Pf(x) + Pf(f(x)))]$	$\neg Pf(f(sk1)) + \neg Pf(f(sk2))$
	$Pf(sk1) + Pf(sk2) + \neg Psk1 + \neg Psk2$
	$Pf(sk1) + \neg Psk1 + \neg Pf(f(sk2))$
	$Pf(sk2) + \neg Psk2 + \neg Pf(f(sk1))$

(The Negated Conclusion CNF is TPTP problem SYN084-2.p.)

4 The Related Problems

Recall that the four problems (Problems 17, 33, 38, 62) are variations on a theme. The theme was set in [2, p. 59]. In discussing their natural deduction system IMPLY, they say:

> IMPLY is incomplete in many ways. For example, although it can prove the skolemized formula
>
> $(P0\&(Px \to Pf(x)) \to Pf(f(x)))$
>
> it cannot handle the following equivalent formula
>
> $(\neg P0 + Px + Pf(f(x)))\&(\neg P0 + \neg Pf(x) + Pf(f(x)))$
>
> because the substitution [0/x] satisfying the first conclusion does not satisfy the second.

Now, it is difficult to know exactly what points the authors intend to make in this quotation, especially with their use of "can prove" and "cannot handle", but at least they seem to be saying that IMPLY would not be able to prove the equivalence of the two formulas. It is exactly this equivalence (adding explicit universal quantifiers and using the constant a rather than 0) that Problem 62 addresses.

The quote also says that the first formula is "skolemized". Again, it is difficult to know what the authors mean by this, since it is not possible to interpret the f's occurring in it as Skolem functions: the $f(f(x))$ would not be interpretable that way. However, it is possible to remove the functions by using equivalent formulas. For example, $Pf(x)$ can be replaced by

(1) $\exists y(Rxy\&\forall z(Rxz \to y = z)\&Py)$

where R is a new predicate not mentioned elsewhere in the problem. [8] does not present a version of Problem 62 using this substitution, but instead gives a 'weaker' version in which the uniqueness of the function is sacrificed by omitting the middle conjunct of (1). Rather than saying that the *unique* thing to which x is R-related is a P, the weaker version says instead that there is something to which x is R-related and that thing is a P. In this scheme, $Pf(x)$ is replaced by

(2) $\exists y(Rxy\&Py)$

It turns out that the effects of the uniqueness presuppositions of the function symbols in Problem 62 are the same on both sides of the equivalence sign in the natural form, so these functions can be uniformly replaced by relations in accordance with (2) and still yield a theorem. This was done in [8] in order to provide a relational predicate logic problem without identity and without function symbols. This is Problem 38 in [8], where only the natural form is given.

Problem 38 Natural FOF

$$(Ax)[(Pa\&(Px \rightarrow (Ey)(Py\&Rxy))) \rightarrow (Ez)(Ew)(Pz\&Rxw\&Rwz)]$$

$$\leftrightarrow$$

$$(Ax)[(\neg Pa + Px + (Ez)(Ew)(Pz\&Rxw\&Rwz))\&$$

$$(\neg Pa + \neg(Ey)(Py\&Rxy) + (Ez)(Ew)(Pz\&Rxw\&Rwz))]$$

The shortest Negated Conclusion CNF we know for Problem 38 contains 46 clauses, and is relegated to the Appendix below.

Further examination of the natural FOF of Problem 38 reveals that the logical contribution of the parts of this formula that say there is something to which x is R-related is the same on each side of the equivalence. Occurrences of these sub-formulas can therefore be uniformly replaced by constants, and the result will still be a theorem. Thus the same problem can be stated as a theorem in monadic predicate logic, which is Problem 33 in [8]:

Problem 33

Natural FOF	**Negated Conclusion CNF**
$(Ax)[Pa\&(Px \rightarrow Pb) \rightarrow Pc]$	Pa
$\leftrightarrow$	$\neg Pa + Px + Pc + Py$
$(Ax)[(\neg Pa + Px + Pc)\&$	$\neg Pa + \neg Pb + Pc$
$(\neg Pa + \neg Pb + Pc)]$	$\neg Pc$
	$Pb + \neg Pd + \neg Pe$
	$\neg Pa + Pb + Pc + Px + \neg Pd$
	$\neg Pa + Pb + Pc + Px + \neg Pe$

(The Negated Conclusion CNF is TPTP problem SYN063-1.p.)

A better clausification is produced by Sutcliffe's clausifier, which reduces the natural form to a trivial Negated Conclusion CNF:

Problem 33 Negated Conclusion CNF

$$Pa$$

$$Pc + \neg Pa$$

$$\neg Pc$$

(These clauses are TPTP problem SYN063-2.p.)

It is well known [1] (see 3, pp. 174–180] for an exposition) that a monadic logic formula is a theorem so long as it is not falsifiable with a domain of size 2^N, where N is the number of distinct predicates (constants being treated as predicates also). This means that monadic logic problems are really propositional logic problems. But *this* monadic logic problem is a particularly easy problem: it is (modulo the translation of constants in Problem 33 to predicates) a substitution instance of the following propositional logic theorem, which is Problem 17 in [8].

Problem 17

Natural FOF	**Negated Conclusion CNF**
$(p\&(q \rightarrow r) \rightarrow s]$	p
$\leftrightarrow$	$\neg p + q + s$
$(\neg p + q + s)$	$\neg p + \neg r + s$
$(\neg p + \neg r + s))$	$\neg s$
	$\neg q + r$

(The Negated Conclusion CNF is TPTP problem SYN047-1.p.)

It therefore seems that the IMPLY system of [2] is unable to solve a simple propositional equivalence.

5 Conclusion

It is interesting to note the varying degrees of difficulty that provers have with Problem 38 (i.e., the relational predicate logic problem without identity and without function symbols). The natural FOF version of Problem 38 is proved quite easily by THINKER. In contrast, OTTER is unable to find a proof of the natural FOF version (which OTTER clausifies to 55 clauses) or of the 46 clause Negated Conclusion CNF, after one hour of DEC station 5000 CPU time, i.e., using fairly large resources. SETHEO [4] is similarly unsuccessful with the 46 clause version. These observations confirm that an essentially easy FOF problem can become a difficult CNF problem.

A Negated Conclusion CNFs for Problem 38

Pa

$Px + Py + Psk1(y) + Psk3(x) + \neg Pa$

$Px + Py + Psk1(y) + Rx, sk4(x) + \neg Pa$

$Px + Py + Psk1(y) + Rsk4(x), sk3(x) + \neg Pa$

$Px + Py + Psk3(x) + Ry, sk2(y) + \neg Pa$

$Px + Py + Rx, sk4(x) + Ry, sk2(y) + \neg Pa$

$Px + Py + Ry, sk2(y) + Rsk4(x), sk3(K) + \neg Pa$

$Px + Psk1(x) + Psk5(y) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Psk1(x) + Ry, sk6(y) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Psk1(x) + Rsk6(y), sk5(y) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Psk5(y) + Rx, sk2(x) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Ry, sk6(y) + Rx, sk2(x) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Rx, sk2(x) + Rsk6(y), sk5(y) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Py + Psk3(x) + Rsk2(y), sk1(y) + \neg Pa$

$Px + Py + Rx, sk4(x) + Rsk2(y), sk1(y) + \neg Pa$

$Px + Py + Rsk2(y), sk1(y) + Rsk4(x), sk3(x) + \neg Pa$

$Px + Psk1(y) + Psk3(x) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Psk3(x) + Ry, sk2(y) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Psk1(y) + Rx, sk4(x) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Rx, sk4(x) + Ry, sk2(y) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Psk3(x) + Rsk2(y), sk1(y) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Rx, sk4(x) + Rsk2(y), sk1(y) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Psk1(y) + Rsk4(x), sk3(x) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Ry, sk2(y) + Rsk4(x), sk3(x) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Rsk2(y), sk1(y) + Rsk4(x), sk3(x) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Psk5(y) + Rsk2(x), sk1(x) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Ry, sk6(y) + Rsk2(x), sk1(x) + \neg Pz + \neg Pa + \neg Ry, z$

$Px + Rsk2(x), sk1(x) + Rsk6(y), sk5(y) + \neg Pz + \neg Pa + \neg Ry, z$

$Psk1(x) + Psk5(y) + \neg Pz + \neg Pu + \neg Pa + \neg Ry, z + \neg Rx, u$

$Psk5(x) + Ry, sk2(y) + \neg Pz + \neg Pu + \neg Pa + \neg Rx, z + \neg Ry, u$

$Psk1(x) + Ry, sk6(y) + \neg Pz + \neg Pu + \neg Pa + \neg Ry, z + \neg Rx, u$

$Rx, sk6(x) + Ry, sk2(y) + \neg Pz + \neg Pu + \neg Pa + \neg Rx, z + \neg Ry, u$

$Psk5(x) + Rsk2(y), sk1(y) + \neg Pz + \neg Pu + \neg Pa + \neg Rx, z + \neg Ry, u$

$Rx, sk6(x) + Rsk2(y), sk1(y) + \neg Pz + \neg Pu + \neg Pa + \neg Rx, z + \neg Ry, u$

$Psk1(x) + Rsk6(y), sk5(y) + \neg Pz + \neg Pu + \neg Pa + \neg Ry, z + \neg Rx, u$

$Rx, sk2(x) + Rsk6(y), sk5(y) + \neg Pz + \neg Pu + \neg Pa + \neg Ry, z + \neg Rx, u$

$Rsk2(x), sk1(x) + Rsk6(y), sk5(y) + \neg Pz + \neg Pu + \neg Pa + \neg Ry, z + \neg Rx, u$

$Psk10 + Psk8 + \neg Psk7 + \neg Psk9$

$Psk8 + Rsk9, sk10 + \neg Psk7 + \neg Psk9$

$Psk10 + Rsk7, sk8 + \neg Psk7 + \neg Psk9$

$Rsk7, sk8 + Rsk9, sk10 + \neg Psk7 + \neg Psk9$

$Psk8 + \neg Px + \neg Psk7 + \neg Ry, x + \neg Rsk9, y$

$Rsk7, sk8 + \neg Px + \neg Psk7 + \neg Ry, x + \neg Rsk9, y$

$Psk10 + \neg Px + \neg Psk9 + \neg Ry, x + \neg Rsk7, y$

$Rsk9, sk10 + \neg Px + \neg Psk9 + \neg Ry, x + \neg Rsk7, y$

$\neg Px + \neg Py + \neg Rz, x + \neg Ru, y + \neg Rsk7, u + \neg Rsk9, z$

(These clauses are TPTP problem SYN067-3.p.)

References

1. P. Bernays and M. Schönfinkel. Zum Entscheidungsproblem der mathematischen Logik. *Mathematische Annalen*, 99:342–372, 1928.

2. W. Bledsoe, R. Boyer, and W. Henneman. Computer proofs of limit theorems. *Artificial Intelligence*, 3:27–60, 1972.

3. D. Kalish, R. Montague, and G. Mar. *Logic: Techniques of Formal Reasoning.* Harcourt Brace Jovanovich, 1980.

4. R. Letz, J. Schumann, S. Bayerl, and W. Bibel. SETHEO: A High-Performance Theorem Prover. *J. Automated Reasoning*, 8(2):183–212, 1992.

5. W. W. McCune. OTTER 3.0 Reference Manual and Guide. Technical Report ANL-94/6, Argonne National Laboratory, 1994.

6. F. J. Pelletier. Completely non-clausal, completely heuristically driven, automated theorem proving. Department Computing Science TR82-7, University of Alberta, 1982.

7. F. J. Pelletier. Thinker. In J. H. Siekmann, editor, *Proceedings of the 8th International Conference on Automated Deduction*:701–702. Springer Verlag, Lecture Notes in Artificial Intelligence 230, 1985.

8. F. J. Pelletier. Seventy-five problems for testing automatic theorem provers. *J. Automated Reasoning*, 2:191–216, 1986.

9. F. J. Pelletier. Errata. *J. Automated Reasoning*, 4:235–236, 1988.

10. J. L. Pollock. *How to Build a Person.* Bradford/MIT Press, 1989.

11. J. L. Pollock. Interest driven suppositional reasoning. *J. Automated Reasoning*, 6(4):419–461, 1990.

12. G. Sutcliffe, C. B. Suttner, and T. Yemenis. The TPTP Problem Library. In A. Bundy, editor, *Proceedings of the 12th International Conference on Automated Deduction*:252–266. Springer Verlag, Lecture Notes in Artificial Intelligence 814, 1994.

13. C. B. Suttner and G. Sutcliffe. The TPTP Problem Library (TPTP v1.2.0). Technical Report AR-95-03 (Technical Report 95/6), Institut für Informatik, Technische Universität München (Department of Computer Science, James Cook University), 1995.